Partially Speaker-Dependent Automatic Speech Recognition Using Deep Neural Networks

Christopher J. Li , Michelle Spigner

Spring Valley High School, 120 Sparkleberry Lane, Columbia, SC, 29229

As people continue to streamline everyday life with technology, the use of speech recognition has been made available in smartphones, speakers, cars, and more. The process of converting audio to text is known as automatic speech recognition (ASR), and current research is focused on improving robustness to speaker variability and background noises. One way to do so is speaker adaptation, which uses a small dataset of speech from one speaker to boost the program's accuracy on the chosen speaker. Due to the literature gap around an accuracy improvement goal for speaker adaptation, this research project aimed to set this goal by directly training ASR programs on the target speaker. It was hypothesized that training the model on data from a specific speaker would improve the model's accuracy when transcribing different speech from the same speaker. Using the Kaldi Toolkit and TIMIT Speech Corpus, speaker-independent and partially speaker-dependent models were trained. Using phone error rate (PER) as a metric for accuracy, it was found that training the model on the target speaker improved mean target speaker accuracy by an absolute PER of 7.4% and mean overall accuracy by 0.5%. Inferential statistics with t-tests revealed that both the increase in target speaker accuracy $t(47.52) = 19.90$, $p < 0.001$ and the increase in overall accuracy $t(57.97) = 3.33$, $p = 0.0015$ were significant. As a result, this experiment presents a successful partially speaker-dependent system that can be used as a goal for novel speaker adaptation approaches.

Introduction

One of the most useful features of popular voice assistants like Apple, Siri, and Amazon Alexa is their ability to process audio and transcribe it into words. This process of converting speech audio into text is known as automatic speech recognition (ASR). Speech recognition has benefits to everyday life because speaking is one of the most natural forms of communication. eMarketer, a market research database, estimated in 2019 that "111.8 million people in the US will use a voice assistant at least monthly".[1] Voice assistants have been integrated into smartphones, smart speakers, cars, TVs, and more. Also, ASR has applications in documenting medical reports, helping handicapped people, processing calls, and preserving endangered languages. The widespread use of speech recognition has resulted in a desire for better transcription accuracy and speed. Current research on ASR focuses on improving its robustness to speaker variability and background noises. One method of doing so is using adaptation algorithms, which are methods of familiarizing baseline ASR systems to a specific source of speech. An investigation of this topic has revealed a significant research gap for identifying the extent that speaker adaptation can improve accuracy. Therefore, this study attempted to use a partially speaker-dependent system to determine the limit of speaker adaptation that novel techniques should strive to achieve.

Speaker-dependent (SD) speech recognition is when the software is trained to recognize one particular speaker while speaker-independent (SI) speech recognition is meant to perform well regardless of the speaker. Although SD systems have much higher accuracy than SI systems, dependent models are limited to only one speaker and require a large amount of training data from the speaker. Shinoda, a professor at the Tokyo Institute of Technology, notes that SD systems struggle when they are asked to transcribe for another speaker.[2] Therefore, this project aimed to improve a SI system's accuracy on a particular speaker while also preserving its ability to transcribe for other unfamiliar speakers. By training a SI ASR program on data from a test speaker, a partially SD system can be built. This system is not fully SD because it should still have the ability to transcribe for speakers it has never been trained on. Two hybrid statistical ASR systems were used to conduct this experiment because hybrid statistical ASR is recent and thoroughly evaluated.

The establishment of a goal for speaker adaptation will notify researchers when their adaptation algorithms are nearing the performance of SD systems. This would help them improve recognition performance especially in situations where people are speaking for a longer time, such as with mobile device assistants, captioning services, and legal transcription. This experiment could also add to the current understanding of other ideas in the field like universal translators, natural language understanding, and text-to-speech programs.

Literature Review

Overview of Speech Recognition

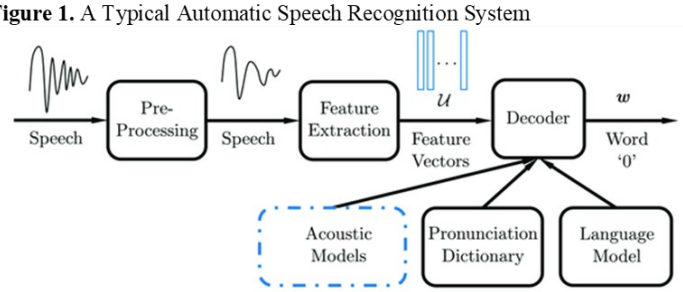
In a presentation affiliated with Microsoft and the Indian Institute of Technology Bombay, Preethi Jyothi explains that ASR is a very challenging task because of the numerous variabilities in speech audio such as background noise, room acoustics, rate of speech, and accent.[3] Although ASR technology has drastically improved from one-word recognition in 1922 to over a million words in the 2000s, it still struggles with variations in age, accent, and noisy backgrounds. In an overview of adaptation techniques, Bell et al. address Jyothi's concern with variation by explaining that "Adaptation algorithms attempt to alleviate the mismatch between the test data and an ASR system's training data." Therefore, adaptation algorithms are valuable because they can help ASR systems with acoustical and speaker differences. One of the first steps to creating the speech recognition system is finding a suitable dataset that can be used for training and testing. Speech datasets are essential for ASR programs because they provide examples of speech data and the corresponding transcriptions for the models to learn from. According to a peer-reviewed paper about speech recognition techniques, ASR software can be categorized as neural network-based, fuzzy logic-based, wavelet-based, and sub-band based.[4] Most of these approaches follow a general format to speech recognition known as statistical speech recognition, which makes use of a deep understanding of speech processing and language-specific statistical models.

Statistical Automatic Speech Recognition

In her presentation, Jyothi explains that the process of statistical ASR begins with acoustic analysis, which involves splitting the audio into 25-millisecond frames and extracting the most influential characteristics from the data. Then, according to the CMUSphinx Project sponsored by Carnegie Mellon University, there are three main models used to output the likely phone and word sequences: the acoustic model, phonetic dictionary, and language model.[5] An acoustic model outputs the most likely phones, the building blocks of words. Gaussian Mixture Models (GMM) and Hidden Markov Models (HMM) are commonly combined in acoustic models. GMMs are statistical classifiers used to match frames

with the phones with the highest probability, and HMMs are probabilistic models that predict a set of unknown variables using a sequence of observed variables. Recently, deep neural networks (DNN) have been combined with HMMs to create a hybrid acoustic model. After the acoustic model, the pronunciation model uses a dictionary to map the phones to the most probable words. The language model restricts word search using n-grams, which show the statistical probabilities of phrases. Lastly, a decoder takes all of this information and searches through a graph of likely phone and word sequences to output the text with the highest probability. A diagram of this pipeline is shown in Figure 1, which was created by Li & Principe,[6] who are researchers from the University of Cambridge and the University of Florida.

Figure 1. A Typical Automatic Speech Recognition System



This diagram outlines the general process of speech recognition, which consists of speech preprocessing, feature extraction, an acoustic model, pronunciation dictionary, language model, and decoder.[6] The outputs from the acoustic, pronunciation, and language models are all utilized in the decoder, which returns the final prediction.

Typically, the performance of an ASR program is evaluated using word error rate (WER), which is calculated as

$$\frac{\text{Substitutions + Insertions + Deletions}}{\text{Number of Words Spoken}}$$

.[5]

Although statistical ASR is generally used in state-of-the-art systems, this architecture can be difficult to adjust because of its various modules and their interactions.

End-to-end Automatic Speech Recognition

In recent years, end-to-end (E2E) systems have emerged as an exciting field. They discard the traditional analytical approach of statistical ASR and instead simplify the entire structure into one neural network that is directly optimized to map acoustic features to characters. This approach is easier to understand and does not require handcrafted models.[3] E2E models are especially promising because they are smaller than hybrid models, and they have more potential to improve. Although E2E models show promise, the hybrid statistical approach was used in this study due to less documentation on E2E models and hardware limitations.

Speech Recognition Adaptation

The purpose of adaptation algorithms is to "require only a few utterances from a user and have as high recognition performance as speaker-dependent systems." [2] There are numerous types of adaptation for speech recognition, such as accent, speaker, domain, and language model adaptation. Speaker adaptation was chosen to be the focus because "Speaker adaptation – adapting the system to a target speaker – is the most common form of adaptation," and it is the easiest to study.[7] Speaker adaptation techniques like maximum a posteriori estimation, maximum likelihood linear regression, and eigenvoice are mentioned in Shinoda's report of adaptation algorithms.[2] At the *International Conference on Acoustics, Speech, and Signal Processing*, Abdel-Hamid and Jiang introduced small speaker codes as a way to quickly adapt DNN-based acoustic models.[8] The speaker codes were fed into a large adaptation network to output a transformation that normalized speaker variation. In a study from the *Journal of Signal Processing Systems*, Xue et al. proposed singular value decomposition (SVD) for adapting hybrid NN/HMM speech recognition models.[9] They reasoned that adjusting singular values slightly would prevent the neural network from overfitting to the adaptation data, which is a common problem with DNNs. Huang and Gong, affiliated with the Microsoft Corporation, presented a method of rapid unsupervised adaptation for hybrid acoustic models.[10] They made use of linear projection layer adaptation, a supervision committee for imperfect supervision, and data augmentation to address data sparsity. Although these acoustic model speaker adaptation techniques focus on bringing SI pipelines closer to the accuracy that SD systems have for particular speakers, there is a lack of research on comparing SD programs and speaker adaptation. The ultimate goal of speaker adaptation is to modify an ASR system so that it is equivalent to training on the target speaker from the beginning. Since this is what the partially SD system does, the accuracy improvement from the partially SD system was expected to be better than that of current speaker adaptation techniques. Therefore, the accuracy of a partially SD system could serve as a goal for ASR experts to use in the future.

Methods

This research project created a partially SD system by adding a specific test speaker to the training set for a hybrid speech recognition system. The goal was to improve the ASR program's accuracy on the target speaker without compromising its accuracy on other speakers. Initially, the plan had been to adapt a pre-trained ASR system using example-weighted neural network training, but this proved to be difficult to implement since it requires a deep understanding of the training process. This adaptation technique was also going to be applied to end-to-end ASR models, but due to time and hardware constraints, this was not accomplished.

The TIMIT Speech Corpus

At the beginning of experimentation, the LibriSpeech Corpus was experimented with, but the 100-hour dataset was too large to use. The Texas Instruments/Massachusetts Institute of Technology (TIMIT) dataset, developed in a joint effort between the Massachusetts Institute of Technology,

Stanford Research Institute International, and Texas Instruments, was chosen for this experiment.[11] It is relatively small and contains speakers with different dialects of English. This speech dataset is a common test for new ASR approaches. For example, Abdel-Hamid and Jiang used the TIMIT dataset to improve upon an adaptation method for convolutional neural network-based models.[8] The TIMIT corpus is composed of 6300 total sentences.[11] Each of the 630 speakers recorded 2 calibration sentences that everyone spoke, 5 phonetically compact sentences that 7 people spoke, and 3 randomly chosen sentences that no one else recorded. The speakers were from 8 dialectical regions of the United States, and about 70% of the speakers were male and 30% were female. The TIMIT dataset was split into training and testing sets. The training set contained 70 to 80% of the speech data while the core test set was utilized for testing, which contained 24 speakers. This training and testing ratio was used because it was already provided for, with useful features like non overlapping speakers and representation of all dialect regions in both sets.

The target speaker was randomly selected from the TIMIT test set, and 5 of the speaker's audio files were designated as training data and the other 5 as testing data. Although previous studies have used a different number of speaker utterances, there were only ten utterances per speaker in the TIMIT dataset. To have effective training and testing for the target speaker, it was decided that five utterances would be used for training and five for testing. In this experiment, the randomly selected speaker had speaker ID "FJLM0". To understand how training on a specific speaker would affect the ASR system's accuracy on different speech from the same speaker, two hybrid speech recognition systems were created. Both were the same, and both had testing datasets with additional target speaker data added. However, one had additional target speaker data in the *training data* while the other did not. Analyzing the difference in accuracy on just the target speaker would show if adding target speaker data to the training dataset improved accuracy on the target speaker. Analyzing the difference in accuracy on the speakers excluding the target speaker would show if adding target speaker data to the training dataset affected overall accuracy. Analyzing target speaker accuracy and overall accuracy was necessary because the goal of this project was to improve the transcription of the target speaker without compromising the ASR system's ability to transcribe for other speakers.

Kaldi Toolkit

To carry out experimentation, the Kaldi toolkit was chosen, which has well over 4,000 citations and was developed by researchers associated with Microsoft Research, Scarland University, the Centre de Recherche Informatique de Montréal, and more.[12] It has been used in numerous studies, such as Miao & Metze's study on speaker adaptation of long short-term memory neural networks.[13] The popularity of the TIMIT dataset and the Kaldi Toolkit in the speech recognition community can be noted because Toledano et al. used both of them to experiment with using a multiresolution representation of the speech signal.[14] Procedures for data preparation, feature extraction, and the training and testing of the hybrid model followed the example scripts and the Kaldi documentation to ensure a decent ASR baseline.[12] Using the Kaldi Toolkit, the standard GMM-HMM model was trained to generate labeled frames for the hybrid DNN component. This process consisted of data preparation, feature extraction, and training the model on the extracted features from the data. Then, the DNN model was trained on the phone to audio alignments from the GMM-HMM system. Finally, the resulting scores were collected using the error rate metric.

Kaldi Installation

Since the Kaldi documentation recommends a Linux environment to compile and run Kaldi, VirtualBox was used to install an Ubuntu Linux virtual machine. Git and Subversion, which are version control systems for organizing code, were installed using the "sudo apt-get" command. Then Kaldi was installed by cloning Kaldi from GitHub, running "extras/check_dependencies.sh" and "make" in the tools directory, and running "./configure", "make depend", and "make" in the src directory. The scripts for training and testing the hybrid model mostly followed the example "run.sh" script found in "egs/timit/s5." The exact script and commands used to run the next steps of the Kaldi pipeline can be found in the "run.sh" script in the GitHub repository in Appendix A.

Data Preparation

To prepare the data using the Kaldi toolkit, data and language folders had to be created. The "data" directory contains the "text", "wav.scp", and "utt2spk" files that direct the program to the speech recordings. The "text" file contains a list of utterance-ids and their respective transcriptions, the "wav.scp" file lists recording-ids and extended filenames, and the "utt2spk" file tells the program which utterance-id corresponds to the speaker-id. The "lang" folder contains information like the lexicon and the phone set. Inside the "lang" directory, the "lexicon" file contains the phone pronunciations for all of the words in the recordings.

Feature Extraction

Feature extraction is the process of reducing the amount of information in a set of data by choosing representative characteristics. Speech frames of 25 ms each shifted by 10 ms were taken to create a discrete representation of the audio. Mel-frequency cepstral coefficients (MFCCs) were used as the acoustic features because they are typically used in other speech recognition programs. MFCCs are inspired by how the human ear works, which makes them effective for identifying the linguistic components of audio and discarding other information.

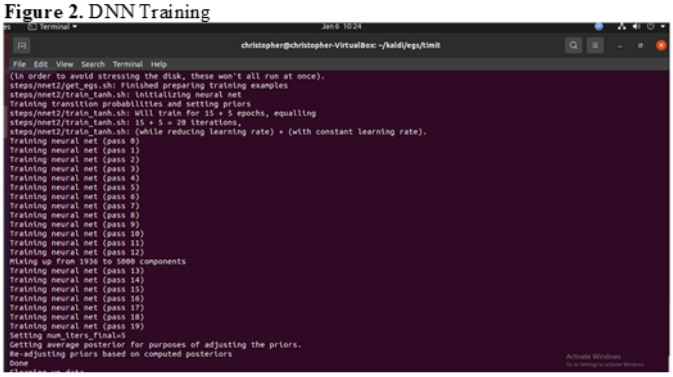
Training Models

In the next 4 stages of the ASR pipeline, the monophone and triphone models were trained. According to Chodroff, a lecturer at the University of York and graduate of Johns Hopkins University, monophone models are acoustic models that have no contextual information about the phones around them, so they are used to train the triphone models, which do use contextual information.[15] After training the monophone model, 3 triphone training algorithms were used to refine the acoustic model. First, delta+delta-delta training was used, which estimates first and second-order derivatives of the features to add to the MFCC features. Then, Linear Discriminant Analysis - Maximum Likelihood Linear Transform (LDA-MLLT) training builds Hidden Markov Model (HMM) states and reduces the differences between speakers. Lastly, Speaker Adaptive Training (SAT) applies speaker and noise normalization to help the model notice the differences between phones rather than the speakers.

The Deep Neural Network Component

The final acoustic model was trained with the deep neural network (DNN), which is a machine learning algorithm used to recognize patterns in data. DNNs contain neurons or nodes, which make up an input layer, hidden layers, and an output layer. These nodes are connected by weights, which determine the importance of each node. Activation functions are then used to determine the output value for each node. Lastly, forward and backward propagation are utilized to train the neural network. In this project, Dan Povey's "nnet2" was used for the hybrid component of the pipeline. It was trained with the tangent hyperbolic function for the activation function. At this point, it was thought that the computer's GPU could be used to speed up the DNN training time, but unfortunately, the CUDA Toolkit used by Kaldi requires an NVIDIA GPU, which was not in the computer. This simply resulted in slower training times. The training process for one of the DNN based acoustic models can be seen in Figure 2.

Figure 2. DNN Training



This screenshot displays the output in the terminal when training one of the DNN acoustic models. It also shows the Linux Ubuntu operating system in the virtual machine on Windows.

Data Collection

After decoding using the "decode.sh" script in the "nnet2" directory, the accuracies of the models were measured with phone error rate (PER), which is similar to WER, but uses phones instead of words. The error rate was calculated as

$$\text{PER} = \frac{\text{Substitutions} + \text{Insertions} + \text{Deletions}}{\text{Number of Phones Spoken}}$$

Substitutions, insertions, and deletions refer to the number of changes required to turn the predicted phone sequence into the correct sequence. To collect the 30 PER scores from each of the 4 sets of data, the "get_scores.py" Python script was written and executed. Lastly, the speech recognition software generated a phone sequence by finding the best path through the phone lattice, which is a graph of phones each associated with a weight that represents their likelihood. A few samples of the correct phone sequence and the program's transcription are shown in Figure 3.

Figure 3. A Sample of the Actual and Predicted Phone Sequences

Sentence ID	Sentence Transcriptions
FDHC0_SI1559	Visually, these approximated what he was feeling within himself.
	sil v ih zh uw ax l iy dh iy z ix cl p r aa cl k s ix m ey dx ix vcl w ah dx iy w ix s f iy l iy n w ih cl th ih n ih m s eh l f sil
	sil v ih sh uw ah l iy dh iy z ih sil p r aa sil k s ih m ey dx ih sil w ah dx iy w ih s f iy l iy n w ih sil th ih n ih m s eh l f sil
FDHC0_SI2189	You saw them always together those years.
	sil y uw s ao dh ah m ao w iy z cl t ix vcl g eh dh er vcl dh ow z y ih er z sil
	sil y uw s aa dh ah m aa w iy z sil t ih sil g eh dh er sil dh ow z y ih er z sil
FDHC0_SI929	But such cases were, in the past, unusual.
	sil b ah cl t s ah cl ch cl k ey s ih z epi w er ix n dh ax cl p ae s cl t ah n y uw zh uw el sil
	sil b ah sil t s ah sil ch sil k ey s ih z sil w er ih n dh ah sil p ae s sil t ah n y uw sh uw l sil

This table shows a sample of the actual and predicted transcriptions from speaker FDHC0. The first row is the original sentence, the second row is the correct phone representation, and the third row is the predicted phone transcription. The "sil" phone stands for silent, so it means there was no phone detected in the frame. The rest of the phones can be sounded out, which roughly correspond to the pronunciation of the original sentence.

Results

Experimentation on hybrid models using the Kaldi Toolkit resulted in 30 trials for each type of system. Due to the non-normal distribution of the phone error rates (PER), the central tendency was represented by median, and spread was measured with interquartile range. The PER of the SI hybrid model on the target speaker (*Median* = 23.5%) was higher than that of the partially SD pipeline (*Median* = 16.8%). The PER of the SI system on all other test speakers (*Median* = 23.7%) was also higher than that of the partially SD system (*Median* = 23.1%). Therefore, training the deep neural network (DNN) on the training data with the target speaker improved median target speaker accuracy by an absolute PER of 6.7% and median overall accuracy by 0.6%. The median and interquartile error rates of the four groups can be seen in Table 1.

Table 1. Phone Error Rates from Speaker-Independent and Speaker-Dependent Hybrid Models

	Target Speaker		Other Test Speakers	
	Median	Interquartile Range	Median	Interquartile Range
Speaker-Independent	**23.5%**	2.7%	23.7%	0.8%
Partially Speaker-Dependent	**16.8%**	0.7%	23.1%	0.9%

This table shows the median and interquartile error rates of the hybrid pipelines. The median PER of the target speaker decreased from 23.5% to 16.8% after training on the chosen speaker, and the median PER of the independent speakers decreased from 23.7% to 23.1% after training on the specific speaker.

To find the statistical significance of the difference between the before and after groups, inferential statistics were conducted. Although histograms of the data did not show a normal distribution, the two-sample t-test was still calculated because it is robust, and it can give valuable insight into the relationships between the scored SI and partially SD datasets.

The objective of this study was to determine whether the partially SD hybrid model would score a lower PER on the target speaker than the SI hybrid model. Therefore, the one tailed two-sample t-test was used to compare the error rates on the target speaker before and after training on the specific speaker. Using the Welch-Satterthwaite equation to calculate degrees of freedom and an alpha level of 0.05, the error rate of the SI system (M = 24.4%, SD = 1.8%) was found to be significantly greater than that of the partially SD model (M = 17.0%, SD = 1.1%) when scored on the target speaker due to $p < 0.001$.

The other objective was to determine whether the overall PER of the hybrid system would be affected by additional training on a target speaker. Since no change was expected, the two tailed two-sample t-test was applied. The overall PER of the SI system (M = 23.8%, SD = 0.6%) was found to be significantly greater than that of the partially SD model (M = 23.3%, SD = 0.6%) because $p = 0.0015$ was lower than the alpha level of 0.05.

Both error rates on the target speaker and the rest of the test set experienced a significant decrease (or increase in accuracy) when the DNN acoustic model was trained on the target speaker's speech in addition to the regular training set. The raw data of 30 trials for each group and the statistical information can be found in Appendix B.

Discussion

This research project aimed to determine how training a DNN based acoustic model on a specific speaker would affect the system's overall accuracy and accuracy in predicting speech from the target speaker. The hypothesis was partially supported because the inferential statistics revealed that the partially SD system did in fact significantly improve (7.4% PER decrease) the accuracy of the ASR program for a specific speaker. However, the hypothesis did not predict that the PER would slightly improve from the SI pipeline to the SD pipeline (0.5% PER decrease). This is still a successful outcome since accuracy improved in both measurements.

When compared to other studies that tested novel speaker adaptation techniques for hybrid ASR programs, this study surpassed them, which was expected because the partially SD model was trained on the target speaker from the beginning. Using small speaker codes on the TIMIT dataset, Abdel-Hamid and Jiang yielded 10% and 5% relative phone error rate reduction using 7 and 1 utterances as adaptation data respectively.[8] With SVD on the Switchboard speech corpus, Xue et al. achieved a 3-6% relative error reduction using a few dozen adaptation utterances.[9] Lastly, Huang & Gong's results showed a 7.3% and 7.9% relative word error rate reduction when they utilized linear projection layer adaptation and 2 to 20 minutes of speaker data.[10] In contrast to these papers, the partially SD system experienced around 28.5% relative phone error rate reduction using 5 sentences from the target speaker. Therefore, adaptation algorithms for hybrid DNN-based models have not yet reached the level of accuracy of a partially SD system.

Conclusions

The results of this study are significant to the study of ASR because it can help experts compare new adaptation techniques with the best possible adaptation, which would be a SD system. Speaker adaptation would be useful in situations where the speaker is using the speech recognition software frequently or for a long time, such as with personalized virtual assistants, court transcription services, and certain Youtube videos. The study also has broader implications in other types of speech adaptation, universal translators, and text to speech programs.

During experimentation, the parameters of the hybrid ASR system were not optimized because the focus was to improve PER after adaptation rather than improve the baseline system. These parameters could be adjusted to increase the accuracy to state-of-the-art levels. Also, only one target speaker was chosen and included in the training data. This study could be expanded upon by checking with many other target speakers to verify that this process will improve the error rate in general. Varying the number of target speaker utterances used for training could be experimented with as well. The hybrid model was limited to the Kaldi pipeline, so future research could make use of other toolkits like the Carnegie Mellon University (CMU) Sphinx Toolkit and Hidden Markov Model Toolkit (HTK). Similarly, other speech datasets like the LibriSpeech ASR Corpus and the Wall Street Journal Corpus could be applied.

This project was limited because it only modeled the effect of adapting a hybrid ASR system to a target speaker. In the future, methods of adapting pretrained DNNs could be investigated to greatly reduce training time. One method that could work with DNNs is example-weighted neural network training, which, according to WolframAlpha, adjusts the weights of the training samples and can be used to assign more importance to specific speakers.[16] Speaker adaptation could also be explored in other popular ASR systems. *Speaker Adaptation for Attention-Based End-to-End Speech Recognition*, in association with the Microsoft Corporation, acknowledges that "there has been limited investigation in speaker adaptation for the E2E ASR." [17] Therefore, future studies could look into adapting E2E speech recognition systems.

Acknowledgements

I would like to thank Mrs. Spigner and Dr. Wyatt for setting deadlines and prompting me to continue with my experiment. I appreciate the support that they provided when I was conducting my experiment because it was a challenging process. I would also like to thank my family for buying my computer.

Notes and References

1. Petrock V. 2019. US Voice Assistant Users 2019. eMarketer. Available from: https://www.emarketer.com/content/us-voice-assistant-users-2019.
2. Shinoda K. 2011. Speaker adaptation techniques for automatic speech recognition. Proc. APSIPA ASC, 2011.
3. Jyothi P. 2017. Automatic Speech Recognition - An Overview. Microsoft Research. [cited 22 November 2020]. Available from: https://www.microsoft.com/en-us/research/video/automatic-speech-recognition-overview/.
4. Haridas AV, Marimuthu R, Sivakumar, VG. 2018. A critical review and analysis on techniques of speech recognition: The road ahead. International Journal of Knowledge-Based and Intelligent Engineering Systems, Vol 22(1): 39–57.
5. Shmyrev N. Basic concepts of speech recognition. CMUSphinx Open Source Speech Recognition. [cited 22 November 2020]. Available from https://cmusphinx.github.io/wiki/tutorialconcepts/.
6. Li K, Principe J. 2018. Automatic speech recognition system diagram. ResearchGate. Available from: https://www.researchgate.net/publication/323737599_Biologically-Inspired_Spike-Based_Automatic_Speech_Recognition_of_Isolated_Digits_Over_a_Reproducing_Kernel_Hilbert_Space.
7. Bell P, Fainberg J, Klejch O, Li J, Renals S, Swietojanski P. 2020. Adaptation Algorithms for Speech Recognition: An Overview. arXiv.org. [cited 22 November 2020]. Available from: https://arxiv.org/abs/2008.06580
8. Abdel-Hamid O, Jiang H. 2013. Fast speaker adaptation of hybrid NN/HMM model for speech recognition based on discriminative learning of speaker code. Proceedings, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp 7942-7946
9. Xue S, Jiang H, Dai L, Liu Q. 2014. Speaker adaptation of hybrid NN/HMM model for speech recognition based on singular value decomposition. Journal of Signal Processing Systems, Vol 82(2), pp 175-185.
10. Huang Y, Gong Y. 2020. Acoustic model adaptation for presentation transcription and intelligent meeting assistant systems. Proceedings, ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
11. Zue V, Seneff S, Glass J. 1990. Speech database development at MIT: Timit and beyond. Speech Communication, Vol 9(4), pp 351–356.
12. Povey D, Ghoshal A, Boulianne G, Burget L, Glembek O, Goel N, Hannemann M, Motlíček P, Qian Y, Schwarz P, Silovsky J, Stemmer G, Vesely K. 2011. The Kaldi speech recognition toolkit. Proceedings, IEEE 2011 workshop on automatic speech recognition and understanding (No. CONF). IEEE Signal Processing Society.
13. Miao Y, Metze F. 2015. On speaker adaptation of long short-term memory recurrent neural networks. Sixteenth Annual Conference of the International Speech Communication Association.
14. Toledano DT, Fernández-Gallego MP, Lozano-Diez A. 2018. Multi-resolution speech analysis for automatic speech recognition using deep neural networks: Experiments on TIMIT. Plos One, Vol 13(10).
15. Chodroff E. 2018. Kaldi Tutorial. [cited 22 November 2020]. Available from: https://eleanorchodroff.com/tutorial/kaldi/index.html.
16. Example-Weighted Neural Network Training. c2021. Wolfram. [cited 22 November 2020] Available from: http://reference.wolframcloud.com/language/tutorial/NeuralNetworksExampleWeighting.html
17. Meng Z, Gaur Y, Li J, Gong Y. 2019. Speaker adaptation for attention-based end-to-end speech recognition. Interspeech 2019. 241-245

Figure 4. Experimental Design Diagram

Title of the Experiment
Partially Speaker-Dependent Automatic Speech Recognition Using Deep Neural Networks

Hypothesis/Engineering Goal
The goal of this project is to imitate speaker adaptation of a hybrid ASR model by adding speech data from a specific speaker to the training dataset. Ideally, the adaptation would improve the accuracy of the system on the target speaker without compromising its ability to transcribe for other speakers. Since training and testing on the same speakers greatly improve performance for statistical models, it was hypothesized that this would also work on hybrid systems.

Independent Variable
The independent variable is whether the system would be trained on the target speaker or not.

Levels of Independent Variable	Partially speaker-dependent model	Speaker-independent model
Number of Trials	30 trials	30 trials

Dependent Variable
The performances of the partially SD and SI models on the target speaker and the other speakers would be measured. Accuracy would be measured with phone error rate (PER), which is calculated as PER = $\frac{Substitutions + Insertions + Deletions}{Number\ of\ Phones\ Spoken}$. (e.g. 20% PER)

Control Group
The control group for the statistical ASR method would be the baseline hybrid ASR program that was trained and tested on the datasets without the target speaker.

Constants
Variables that would remain constant between the SI (baseline) and SD systems are

- the speech preprocessing
- the hybrid ASR pipeline
- the characteristic that is optimized (PER)
- the output format
- a large portion of the training and testing datasets

This table displays a general outline of the experiment including the title, hypothesis, independent variable, dependent variable, control group, and constants.

Appendix

A. Project Files
https://tinyurl.com/ASRGitHub
This is the GitHub repository for this project. It contains the data, scripts, and ASR models that were used in the experiment.

B. Raw Data
https://tinyurl.com/RawPERData
This is a Google spreadsheet containing the 4 sets of data each containing 30 scores. It also shows the results of the descriptive and inferential statistics used to analyze the data.